IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Currently Amended): A data access, replication or communications system comprising: ~~a software application that is distributed across a terminal-side component running on a terminal and a server-side component;~~

~~in which the terminal-side component and the server-side component (i) together constitute a client to a server and (ii) collaborate by sending messages using a message queuing system over a network.~~

a terminal including an electronic memory storing a terminal-side executable and a processor provided to execute the terminal-side executable to enable communication therewith independent of a session-based protocol; and

a server including an electronic memory storing a server-side executable and a processor provided to execute the server-side executable to enable communication therewith independent of a session-based protocol,

wherein, the terminal-side executable and the server-side executable exchange messages using a message queuing system over a network and cooperatively function as a client of a second server.

Claim 2 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the message queuing system is message oriented middleware.

Claim 3 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable insulates a terminal program from being affected if [[the]] a connection over the network is broken by [[also]] queuing messages in readiness for the connection to be re-established, enabling the terminal program to proceed to ~~its next~~ another task.

Claim 4 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the server-side ~~component~~ executable insulates a server program from being affected if [[the]] a connection over the network is broken by [[also]] queuing messages in readiness for the connection to be re-established, enabling the server program to proceed to ~~its next~~ another task.

Claim 5 (Currently Amended): The system of Claim 1 ~~in which~~ wherein each message that is queued defines part or all of an event, ~~in which an~~ the event ~~describes~~ describing a change to [[the]] data stored at either the terminal or server in enough detail to enable data replication to take place without [[the]] a need for a synchronization ~~synchronisation~~ engine[[;]], data replication being achieved by sending events rather than a complete dataset (or sub-sets of a dataset) of stored data for synchronization ~~synchronisation~~.

Claim 6 (Currently Amended): The system of Claim 5 ~~in which~~ wherein the terminal-side ~~component~~ executable can create and queue events ~~and queue those events, itself and/or in the message queuing system~~, enabling the terminal-side ~~component~~ executable to proceed to ~~its next~~ another task, even if [[the]] a network connection is broken, the events being queued in one of the terminal-side executable and a message queuing system.

Claim 7 (Currently Amended): The system of Claim 5 ~~in which~~ wherein the server-side ~~component~~ executable can create and queue events ~~and queue those events, itself and/or in the message queuing system~~, enabling the server-side ~~component~~ executable to proceed to

~~its next~~ another task, even if [[the]] a network connection is broken, the events being queued in one of the server-side executable and a message queuing system.

Claim 8 (Currently Amended): The system of Claim 6 ~~in which~~ wherein the queued events persist in non-volatile memory [[even]] when the terminal is switched off.

Claim 9 (Currently Amended): The system of Claim 7 ~~in which~~ wherein queued events persist in non-volatile memory [[even]] when the server is switched off.

Claim 10 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable and the server-side ~~component collectively constitute~~ executable cooperatively function as middleware between a terminal program running on [[the]] a wireless terminal and a server program running on the server.

Claim 11 (Currently Amended): The system of Claim 6 ~~in which~~ wherein messages ~~that are~~ queued on the terminal side are references to data stored ~~held~~ on the server.

Claim 12 (Currently Amended): The system of Claim 10 ~~in which~~ wherein a message queuing system on the terminal side insulates the terminal program from being affected if [[the]] a connection over the network is re-established by automatically causing [[the]] a next message in a terminal-side queue to be sent.

Claim 13 (Currently Amended): The system of Claim 10 ~~in which~~ wherein a message queuing system on the server side insulates the server program from being affected if [[the]] a connection over the network is re-established by automatically causing [[the]] a next message in a server-side queue to be sent.

Claim 14 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable processes events from a terminal program, which is an e-mail or PIM program.

Claim 15 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the server-side ~~component~~ executable processes events from a server program, which is a mail server program.

Claim 16 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal is a wireless terminal such as a mobile telephone or smartphone.

Claim 17 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the network is a wireless WAN network such as a GPRS or UMTS network.

Claim 18 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the server-side ~~component~~ executable stores a logon password sent from the terminal-side ~~component~~ executable and can use [[this]] the logon password to access a server program.

Claim 19 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the server-side ~~component can assemble a message that the terminal-side component wishes to send by using data held on the server in order to avoid that data needing to be sent over the network~~

~~from the terminal~~ executable stores data on the server to assemble a sent message, to which fewer than all data was received from the terminal.

Claim 20 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable monitors available memory on the terminal and automatically deletes data stored on the terminal that meets a pre-defined criteria of at least one of age, ~~and/or~~ use ~~and/or~~ and size without affecting [[the]] a corresponding data stored on the ~~terminal~~ server.

Claim 21 (Currently Amended): The system of Claim 20 ~~in which~~ wherein a user option to delete data stored on the terminal without affecting the corresponding data stored on the server is displayed at [[the]] a same level in a menu hierarchy, displayed on the terminal, as an option to delete data stored on the terminal together with the corresponding data stored on the server.

Claim 22 (Currently Amended): The system of Claim 20 ~~in which~~ wherein the data is message data and the terminal side ~~component~~ executable retains data that allows the message data to be re-supplied from the server if requested by a user.

Claim 23 (Currently Amended): The system of Claim 20 ~~in which~~ wherein data is not released from memory if the data is marked as unread, open for user viewing or action, or there is a pending action related to the data requesting additional data from the ~~large~~ second server.

Claim 24 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable enables a document attachment to be sent to the wireless terminal in either [[the]] an original format in which the document is stored [[at]] on the server or in ~~a more~~ another useable format converted from the original format.

Claim 25 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable enables a user to [[(a)]] select a release option to delete a message stored on the terminal ~~but not the~~ without deleting a corresponding message stored on the server and [[also]] to [[(b)]] select a delete option to delete a message stored on the terminal and [[also]] the corresponding message on the server, the release and delete options appearing at [[the]] a same level in a menu hierarchy displayed on the terminal.

Claim 26 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the client, implemented by the terminal-side and server-side executables, ~~application~~ enables [[the]] a correct routing of messages addressed to a terminal identified by an ID by mapping that ID to [[the]] an actual IP address needed to reach the terminal.

Claim 27 (Currently Amended): The system of Claim 26 ~~in which the~~ wherein an address is a dynamic IP address allocated by a NAT box.

Claim 28 (Currently Amended): The system of Claim 27 ~~in which~~ wherein the ~~application~~ client, implemented by the terminal-side and server-side executables, only initiates a message transfer if there exists a valid mapping.

Claim 29 (Currently Amended): The system of Claim 28 ~~in which~~ wherein a mapping is refreshed whenever a specific kind of ~~small~~, dedicated message having fewer than 20 bytes is received from the terminal.

Claim 30 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the terminal-side ~~component~~ executable allows a server administrator to lock an application on the terminal without affecting other applications on the terminal.

Claim 31 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the ~~terminal component~~ terminal-side executable sends a challenge to any third party ~~suspected of~~ identified as attempting a denial of service attack on the terminal, preventing ~~and that denial of service attack does not then lead to~~ any additional data traffic to the terminal from the denial of service attack.

Claim 32 (Currently Amended): The system of Claim 1 ~~in which~~ wherein the ~~application comprises~~ client, implemented by the terminal-side and server-side executables, includes a distributed application platform that makes calls to a distributed communications platform.

Claim 33 (Currently Amended): The system of Claim 32 ~~in which the~~ wherein the distributed communications platform enables reliable delivery of a message over the network ~~to be reliable~~, ~~even if an~~ independently of any unreliable transport protocol [[is]] used. ~~, in which the platform operates in a session independent manner.~~

Claim 34 (Currently Amended): A method of data access, replication or communication comprising the steps of:

~~(a)~~ running ~~a software application that is distributed across~~ a terminal-side ~~component~~ executable on a terminal to enable communication with a server independent of a session-based protocol;

running a server-side executable on the server to enable communication with a terminal independent of a session-based protocol, the terminal-side executable and the server-side executable cooperatively functioning as a client to a second server; and ~~a server-side component, in which the terminal-side component and the server-side component together constitute a client to a server;~~

~~(b)~~ sending messages between the terminal-side executable ~~component~~ and the server-side ~~component~~ executable using a message queuing system over a network.

Claim 35 (Currently Amended): The method of Claim 34, wherein

the message queuing system is message oriented middleware,

the terminal-side executable insulates the terminal from being affected by a broken network connection, enabling the terminal to proceed to another task, the terminal-side executable queuing messages in readiness for the network connection to be re-established, and

the server-side executable insulates the server from being affected by a broken network connection, enabling the server to proceed to another task, the server-side executable queuing messages in readiness for the network connection to be re-established. ~~in which the software application is an element of a system defined in any preceding Claim 1-33.~~

Claim 36 (Currently Amended): A terminal comprising:

an electronic memory configured to store a terminal-side executable to enable communication therewith independent of a session-based protocol;

a processor configured to execute the terminal-side executable to

insulate the terminal from a broken network connection, enabling the terminal to proceed to another task,

queue messages in readiness for the network connection to be re-established, the messages being at least a part of an event describing a change to data stored on a server and allowing data replication without transmission of an entire dataset, and

automatically sending the queued messages upon re-establishment of the network connection via at least one of a wired connection or a wireless connection to the network.

~~when programmed with the terminal-side component that is an element of a system as defined in any preceding Claim 1-33.~~

Claim 37 (Currently Amended): A server comprising:

an electronic memory configured to store a server-side executable to enable communication therewith independent of a session-based protocol;

a processor configured to execute the server-side executable to

insulate the server from a broken network connection, enabling the server to proceed to another task,

queue messages in readiness for the network connection to be re-established, the messages being at least a part of an event describing a change to data stored on a terminal and allowing data replication without transmission of an entire dataset, and

automatically sending the queued messages upon re-establishment of the network connection via at least one of a wired connection or a wireless connection to the network.

~~when programmed with the server-side component that is an element of a system as defined in any preceding Claim 1-33.~~